Algorithme MCMC

11 novembre 2015

I. Introduction

definition

Un algorithme *Markov Chain Monte Carlo* (MCMC) est un algorithme stochastique qui permet de simuler une distribution à l'aide d'une chaîne de Markov.

definition

Un algorithme *Markov Chain Monte Carlo* (MCMC) est un algorithme stochastique qui permet de simuler une distribution à l'aide d'une chaîne de Markov.

1. Pourquoi et comment simuler ? Premiers algorithmes stochastiques.
2. Chaînes de Markov
3. MCMC par l'approche de Metropolis-Hastings
4. MCMC par échantillonneur de Gibbs

I. Pourquoi et comment simuler ? Premiers algorithmes stochastiques

I.1 Calculs d'intégrales

Problème

Déterminer

$$I = \int_a^b h(t)\,dt$$

- Calcul d'intégrale
- Méthode des trapèzes

$$\hat{I} = \sum_{i=1}^{n-1} (x_{i+1} - x_i) \frac{h(x_i) + h(x_{i+1})}{2}$$

- Méthode des splines
- ...

On reécrit le problème sous la forme suivante :

Problème

Déterminer

$$I = \int_{Omega} h(t)f(t)dt$$

où f désigne une densité sur l'espace Ω.

- $\Omega = [a, b]$
- La nouvelle fonction h es l'ancienne fonction h divisée par f
- Le choix de f est libre sous la condition $\Omega \subset supp(f)$

Idée clé

On a reécrit I sous la forme

$$I = \mathbb{E}_f(h(X))$$

Rappel

Loi forte des grands nombres

Soient $X_1, X_2, \cdots, X_n$ des variables aléatoires de même loi qu'une variable aléatoire X.
Alors, presque sûrement (c'est-à-dire avec probabilité 1),

$$\lim_{n \to +\infty} \frac{X_1 + \ldots + X_n}{n} = \mathbb{E}X$$

TCL

Soient $X_1, \cdots, X_n$ des variables aléatoires indépendantes et identiquement distribuées d'espérance μ et de variance σ^2. On note $\overline{X}_n = n^{-1} \sum_{i=1}^{n} X_i$. Alors la loi de $\frac{\overline{X}_n - \mu}{\sigma / \sqrt{n}}$ tend vers la loi normale centrée réduite.
En d'autre termes, pour tous a et b réels,

$$lim_{n \to \infty} \mathbb{P}\left[a \leq \sqrt{n} \left(\frac{\overline{Y}_n - \mu}{\sigma} \right) \leq b \right] = \mathbb{P}(a \leq Z \leq b)$$

où Z est une variable gaussienne centrée réduite, $Z \sim \mathcal{N}(0, 1)$.

Remarque : Ce résultat reste vrai quand σ est remplacé par $\widehat{\sigma}$, un estimateur consistant de σ.

Méthode de Monte-Carlo pour le calcul d'intégrale

Idée clé

On a reécrit I sous la forme

$$I = \mathbb{E}_f(h(X))$$

Soit $(x_i)_{1 \leq i \leq n}$ un échantillon simulées suivant la distribution f et

$$\overline{h_n} = \frac{1}{n} \sum_{i=1}^{n} h(x_j)$$

- La loi forte des grands nombres certifie alors que, presque sûrement, $\lim_{n \to \infty} \overline{h_n} = \mathbb{E}_f(h(X)) = I$.

Méthode de Monte-Carlo pour le calcul d'intégrale

- Le TCL permet d'estimer l'erreur commise :

$$\mathrm{Var}(\overline{h_n}) = \frac{1}{n}\mathrm{Var}(h(X)) = \frac{1}{n}\int_\Omega (h(t) - \mathbb{E}_f(h(X)))^2 f(t)\,dt$$

  Cette variance peut être estimée de façon consistante par

$$v_n = \frac{1}{n^2}\sum_{i=1}^{n}(h(x_j) - h_n)^2$$

  .

Remarque : La méthode se généralise au cas des distributions discrètes (en remplaçant les intégrales par des sommes) et au cas des distributions à plusieurs dimensions (en considérant des intégrales multiples).

Echantillonnage préférentiel

Soit g une densité définie sur Ω telle que $supp(h \times f) \subset supp(g)$.

$$\mathbb{E}_f(h(X)) = \int_\Omega \frac{h(t)f(t)}{g(t)} g(t)\,dt = \mathbb{E}_g(\frac{h(X)f(X)}{g(X)}) \tag{1}$$

La méthode de Monte-Carlo peut alors être appliquée en échantillonant suivant g plutôt que suivant f et en approchant l'intégrale par

$$\frac{1}{n}\sum_{i=1}^{n} \frac{h(x_j)f(x_j)}{g(x_j)}$$

La convergence vers $\mathbb{E}_f(h(X))$ quand n tend vers l'infini reste vraie, la différence étant que la variance de l'estimateur est alors

$$\frac{1}{n}\int_\Omega (\frac{h(t)f(t)}{g(t)} - \mathbb{E}_f(h(X)))^2 g(t)\,dt \tag{2}$$

Un choix judicieux de g peut réduire cette variance et donc l'amplitude des intervalles de confiance.

Echantillonnage préférentiel

Choix de g

Prendre une fonction qui échantillonne préférentiellement dans les régions où fh est élevé et telle que $\int_\Omega \frac{f^2(t)h^2(t)}{g(t)}\,dt$ converge (ce qui revient à dire que la variance de l'estimateur existe).

Echantillonnage préférentiel

Choix de g

Prendre une fonction qui échantillonne préférentiellement dans les régions où fh est élevé et telle que $\int_\Omega \frac{f^2(t)h^2(t)}{g(t)}\,dt$ converge (ce qui revient à dire que la variance de l'estimateur existe).

Exemple : On cherche à déterminer la p-valeur $\mathbb{P}(Z > 4)$ quand Z suit une loi normale centrée réduite.

- f la densité d'une loi normale centrée réduite et $h(t) = \mathbb{I}_{t>4}$. La méthode de Monte-Carlo appliquée à h et f va dans ce cas se révéler très lente puisque l'énorme majorité des valeurs échantillonnées suivant $\mathcal{N}(0,1)$ vont être inférieures à 4 (la vraie valeur recherchée étant de $3.2 \quad 10^{-5}$, à peu près 1 valeur sur 30000 sera non nulle).
- Echantillonnage préférentiel avec g la densité d'une loi exponentielle de paramètre $\frac{1}{4}$. La proportion de valeurs échantillonées non nulle passe alors à plus d'un tiers, accélerant la convergence de l'algorithme.

Pour et contre

Avantages des méthodes numériques

- Elles tiennent en compte la forme de la fonction h ;
- Elles sont plus rapides pour les fonctions régulières et en petite dimension.

Avantage de l'approche stochastique

- Elle ne passe pas beaucoup de temps à gérer les difficultés de régularité dans les zones de faible probabilités ;

I.2 Optimisation - Estimation

Estimation paramétrique fréquentiste

On dispose d'un échantillon $\mathbf{x} = (x_1, \ldots, x_n)$ de données.

- on fait l'hypothèse que les observations suivent une loi connue qui dépend d'un vecteur de paramètres $\theta = (\theta_1, \ldots, \theta_p)$.
- on fait l'hypothèse que les observations sont indépedantes et on en déduit la vraisemblance $\mathcal{L}$ de l'observation.
- on estime les paramètres en maximisant la vraisemblance.

Remarque : Quitte à remplacer le fonction à optimiser par son opposé, maximiser et minimiser une fonction sont le même problème.

Méthode 1 : Résolution analytique

- La vraisemblance est une fonction qu'on sait maximiser en plusieurs variables et on obtient des formules closes d'estimation.

Exemple : On suppose que les tirages X_i suivent une loi normale $\mathcal{N}(\mu, \sigma^2)$. La log-vraisemblance de l'échantillon $\mathbf{x} = (x_1, \ldots, x_n)$ vaut alors

$$\log \mathcal{L} = -\frac{n}{2} \log(2\pi\sigma) + \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{\sigma^2}$$

et les valeurs maximisant cette fonction à deux variables sont $\hat{\mu} = \overline{\mathbf{x}}$ et $\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{\mathbf{x}})^2$.

Données censurées

La vraisemblance peut cependant être trop compliquée pour être optimisée analytiquement.

- Si la variable qu'on mesure est le minimum entre plusieurs variables. Par exemple dans des études de survie, où plusieurs raisons peuvent faire sortir le patient de la cohorte.

Exemple $X = min(X_1, X_2)$ avec $X_1 \sim \mu_\infty, \sigma_\infty^\in$ et $X_2 \sim \mu_\in, \sigma_\in^\in$.

$$f_Y(x) = \big(1 - \Phi(\frac{x-\mu_1}{\sigma_1})\big)\frac{1}{\sigma_2}\phi(\frac{x-\mu_2}{\sigma_2}) + \big(1 - \Phi(\frac{x-\mu_1}{\sigma_1})\big)\frac{1}{\sigma_2}\phi(\frac{x-\mu_2}{\sigma_2})$$

Méthode 2 : Descente de gradient

On choisit un point de départ x_0 et on construit une suite de points θ_i suivant la relation de récurrence

$$x_{i+1} = x_i - \alpha_i \nabla h(x_i)$$

où (α_i) est une suite positive tendant vers 0.

- Converge vers un minimum local
- Converge vers le minimum global si la fonction est convexe

Distributions multimodales

- La vraisemblance peut avoir un grand nombre de maxima locaux, et il faut tous les déterminer pour trouver le maximum global.

Exemple : Une alternative à la loi normale est la loi de Student de loi $\mathcal{T}(p, \mu, \sigma)$.
Sa densité est $t(x) = \frac{1}{\sigma}\left(1 + \frac{(x-\mu)^2}{p\sigma^2}\right)^{-\frac{p+1}{2}}$.
Pour p connu et (μ, σ) inconnus, la vraisemblance

$$\mathcal{L} = \frac{1}{\sigma^n} \prod_{i=1}^{n} \left(1 + \frac{(x_i - \mu)^2}{p\sigma^2}\right)^{-\frac{p+1}{2}}$$

a n maxima locaux.

Modèles de mélanges

- Si le modèle comprend des variables cachées, la vraisemblance contient un nombre exponentiel de termes
- La vraisemblance est possiblement multimodale
- Dans ce cas, l'algorithme EM est une alternative (cf algo stat)

Exemple Modèle de mélange à K classes : chaque individu appartient à la classe i avec probabilité α_i. La distribution de X dans la classe α_i est f_i.

$$\mathcal{L}(\mathbf{x}) = \prod_{j=1}^{n} \left(\sum_{i=1}^{K} \alpha_i f_i(x_j) \right)$$

Développer cette vraisemblance amènerait à K^n termes.

Méthode 3 : Monte-Carlo pour l'optimisation

Soit h la fonction à maximiser.

- On simule une suite de valeurs $(x_i)_{1 \leq i \leq n}$ suivant une loi f et on renvoie $\max h(x_i)$.
- Choisir f indépendamment de h risque d'entraîner une convergence très lente car on risque de devoir simuler très longtemps avoir de tirer des points suffisamment proches des endroits où h prend son maximum.
- Une famille de loi couramment utilisée dans ce but est de de choisir $H(\theta) \propto exp(h(\theta)/T)$ pour $T > 0$ (le paramètre T est appelé température).
  Une distribution de ce type prend son maximum au même endroit que h.
  De plus, quand la température tend vers 0, la distribution H se concentre autour des points où le maximum est atteint.

Remarque : Nécessite de pouvoir simuler suivant une loi qu'on ne connaît qu'à une constante multiplicative près.

I.3 Statistiques bayésiennes

Statistiques bayésiennes : Idée générale

- Approche différente de l'approche fréquentiste : les paramètres θ sont considérées des variables aléatoires.
- On munit $\boldsymbol{\theta}$ d'une loi *à priori* $\mathbb{P}(\theta)$, ne dépendant pas des données. On peut la choisir non-informative ou au contraire y injecter des connaissances à priori sur le problème.
- On définit une loi des observations étant donné les paramètres $\mathbb{P}(\mathbf{x}|\theta)$, comme dans le cas fréquentiste.
- On utilise la formule de Bayes

$$\mathbb{P}(\theta|\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x}|\theta)\mathbb{P}(\theta)}{\mathbb{P}(\mathbf{x})} \tag{3}$$

- On en déduit la loi loi *à posteriori* $\mathbb{P}(\theta|\mathbf{x})$. Elle correspond à la vision de la loi de $\boldsymbol{\theta}$ après qu'on ait vu les données.

Avantages

- Il est possible d'intégrer des connaissances autres que celles de l'observation $\mathbf{x}$ dans la loi à priori.
- Le résultat pour θ étant une loi et non pas une valeur, on obtient aisément des intervalles de confiance en considérant les quantiles adéquats.

Statistiques bayésiennes : exemple (inspiré de Dobson et Barnett)

On considère qu'un village est touché de façon endémique par un ver parasitaire (*Schistosoma japanicum*) si plus de la moitié du village est infecté.
Soit θ la proportion de villageois touchés.
On examine 10 personnes, dont 7 sont touchées. On a alors la vraisemblance $\mathbb{P}(x|\theta) = \binom{10}{7}\theta^7(1-\theta)^3$.

Cas 1 : pas d'à priori

Si on a aucun à-priori sur la valeur de θ, on choisit le distribution uniforme $\mathcal{U}[0,1]$.
On obtient la loi à postériori

$$\mathbb{P}(\theta|x) \propto \theta^7(1-\theta)^3$$

Le résultat en terme d'interprétation et d'intervalle de confiance est très proche du l'intervalle de confiance fréquentiste.

Statistiques bayésiennes : exemple (inspiré de Dobson et Barnett)

On considère qu'un village est touché de façon endémique par un ver parasitaire (*Schistosoma japanicum*) si plus de la moitié du village est infecté.
Soit θ la proportion de villageois touchés.
On examine 10 personnes, dont 7 sont touchées. On a alors la vraisemblance $\mathbb{P}(x|\theta) = \binom{10}{7}\theta^7(1-\theta)^3$.

Statistiques bayésiennes : exemple (inspiré de Dobson et Barnett)

On considère qu'un village est touché de façon endémique par un ver parasitaire (*Schistosoma japanicum*) si plus de la moitié du village est infecté. Soit θ la proportion de villageois touchés.
On examine 10 personnes, dont 7 sont touchées. On a alors la vraisemblance $\mathbb{P}(x|\theta) = \binom{10}{7}\theta^7(1-\theta)^3$.

Cas 2 : un à-priori défavorable

Des données autres (salubrité, accès à l'eau, aux sons...) nous font penser qu'il y a une plus grande chance qu'il y ait beaucoup d'infectés. On choisit par exemple une loi à-priori de densité 2θ.
On obtient alors la loi à postériori

$$\mathbb{P}(\theta|x) \propto \theta^8(1-\theta)^3$$

Le résultat en terme d'interprétation diffère maintenant du cas fréquentiste puisque la valeur θ de plus grande probabilité à posteriori est maintenant $\frac{8}{11} > \frac{7}{10}$. Cette différence s'accroit évidemment si la distribution à priori penche encore plus fortement vers les grandes valeurs.

Hyperparamètres

Remarques

- Si les lois à priori et à postériori dépendent de paramètres, on les appellent des *hyperparamètres*.
- Pour une forme vraisemblance donnée, il existe parfois une forme fonctionnelle pour la loi à priori telle que la loi à postériori est de la même famille fonctionnelle. On parle alors de *loi conjuguée*. Par exemple, pour une vraisemblance binomiale, une à priori en loi Beta donnera une postérieure en loi Beta.
  Inférer la loi à posteriori revient alors à déterminer les hyperparamètres.

Exemple : Dans l'exemple précédent, où $p(x|\theta)$ suit une loi binomiale, on sait que si la loi à priori est une loi Beta, la loi à posteriori sera également une loi Beta. On peut par exemple mettre en place une procédure du type :

- partir d'une distribution non-informative $Beta(1, 1)$.
- faire des premières mesures et obtenir une distribution $Beta(a_1, b_1)$.
- Si de nouvelles mesures sont disponibles, partir de l'à-priori $Beta(a_1, b_1)$ et obtenir une nouvelle distribution $Beta(a_2, b_2)$
- ...

Pourquoi simuler en bayésien ?

$$\mathbb{P}(\theta|\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x}|\theta)\mathbb{P}(\theta)}{\mathbb{P}(\mathbf{x})} \tag{4}$$

Le dénominateur ne peut en général pas se déterminer autrement que par

$$\mathbb{P}(\mathbf{x}) = \int \mathbb{P}(\mathbf{x}|\theta)\mathbb{P}(\theta)\, d\,\boldsymbol{\theta}$$

(en remplaçant $\int d\,\boldsymbol{\theta}$ par $\sum_{\boldsymbol{\theta}}$ le cas échéant).
Cette quantité ne peut pas être calculée dans la plupart des cas.

Intervalle de confiance

- Savoir simuler sous la loi à postériori permet de déterminer des intervalles de confiance pour θ.

Exemple : Considérons une régression logistique où la variable observée vérifie

$$\mathbb{P}(Y = 1) = \frac{exp(\mathbf{x}^t \boldsymbol{\theta})}{1 + exp(\mathbf{x}^t \boldsymbol{\theta})}$$

Obtenir un intervalle de confiance pour $\boldsymbol{\theta}$ ne peut être fait de façon théorique. Considérer l'approche bayésienne puis simuler sous la loi à posteriori $p(\boldsymbol{\theta} \,|\, \mathbf{x}, \mathbf{y})$ est une manière d'y arriver.

- Il faut être capable de simuler sous une loi connue à une constante près.

I.4 Premiers algorithmes de simulation

Distribution uniforme $\mathcal{U}[0,1]$

- Il n'est pas possible de produire une liste de nombre complètement aléatoire. On produit forcément une suite $(u_n)_{n\in\mathbb{N}}$
- Algorithmes de congruence : $x_{n+1} = Ax_n + B[M]$ et $u_n = \frac{x_n}{M}$
- Algorithmes de déplacement de registre : $x_n \in [0, 2^{k-1}]$ est représenté par le vecteur $X_n \in \{0,1\}$ de sa séquence de bits. $X_{N+1} = AX_N$ où A est une matrice binaire fixée et $u_{n+1} = \frac{x_{n+1}}{2^k}$
- Ces deux méthodes donnent des suites périodiques. L'algorithme KISS qui alterne les deux permet d'obtenir une période de 2^{95}.
- Toute une batterie de tests (Kolmogorov-Smirnov, DieHard, ...) ne permettent pas d'invalider que la suite obtenue est distribuée uniformément.

Distribution quelconque de fonction de répartition connue

Propriété

Soit F la fonction de répartition de X. On considère l'inverse généralisé de F

$$F^{-}(u) = inf\{x; F(x) \geq u\}$$

Alors, si U suit une loi uniforme sur $[0, 1]$, $F^{-}(U)$ suit la loi de X.

Exemple : Si $X \sim \mathcal{E}(1)$, on a $F(x) = 1 - e^{-x}$. Son inverse est $F^{-}(u) = -log(1 - u)$.
Par conséquent, si U suit une loi uniforme sur $[0, 1]$, $-logU$ suit une loi $\mathcal{E}(1)$

Transformation d'une loi simulable

Theoreme de Box-Muller

Soit R et Θ des variables aléatoires telles que $R \sim \mathcal{E}(\frac{1}{2}$ et $\Theta \sim \mathcal{U}[0,1]$. Alors $X = \sqrt{R}\cos(2\pi\Theta)$ et $Y = \sqrt{R}\sin(2\pi\Theta)$ sont indépendantes et de loi $\mathcal{N}(0,1)$.

- Problème : manque de généralité, il faut déterminer une méthode par distribution

Méthode d'acceptation-rejet

Soit f la densité sous laquelle on cherche à simuler, appelée *densité cible*. On considère une autre densité g, appelé *densité instrumentale*, telle que :

- il est aisé de simuler suivant g
- $supp(f) \subset supp(g)$
- il existe une constante M telle que $f(x) \leq Mg(x)$ pour tout x.

Algorithme d'acceptation-rejet

1. Générer Y suivant la loi g.
2. Générer U suivant une loi $\mathcal{U}[0,1]$
3. Accepter (c'est-à-dire ajouter à l'échantillon) la valeur Y si $U < \frac{f(Y)}{Mg(Y)}$

Méthode d'acceptation-rejet

Propriété

L'échantillon généré par la méthode précédente suit la loi de X.

- Le choix de M est important. Si M est choisi trop grand, l'algorithme prend plus de temps à converger.
- Dans certaines situations (stat bayésiennes par exemple), f n'est connue qu'à une constante près : comment choisir M ?

Simulation vs Méthode déterministes

Avantages de la simulation

- pas besoin de connaitre la forme de la distribution
- s'adapte aux distributions mutimodales
- on passe peu de temps sur les zones de faible probabilité, qui impacte peu le résultat

Avantages du déterminisme

- on tire parti de la forme de la distribution
- plus précis et efficace sur les distributions lisses/ unimodales/ en faible dimension

II. Chaînes de Markov

Définition

Chaîne de Markov

Une suite $(X_i)_{i\geq 0}$ de v.a. discrètes est appelée *chaîne de Markov* si elle vérifie la *propriété de Markov*, qui caractérise les processus sans mémoire :

$$\mathbb{P}(X_{i+1} = x_{i+1}|X_i = x_i, X_{i-1} = x_{i-1}, \ldots, X_0 = x_0) = \mathbb{P}(X_{i+1} = x_{i+1}|X_i = x_i)$$

On note π_i la distribution de X_i. La chaîne est alors caractérisée de façon unique par la distribution π_0 et par ses *probabilités de transition* $((p_{qr}))_{q,r\in\mathcal{S}^2}$ entre états

$$p_{qr} = \mathbb{P}(X_{i+1} = r|X_i = q)$$

La matrice P (éventuellement infinie si $\mathcal{S}$ est un ensemble dénombrable) regroupant les $((p_{qr}))_{q,r\in\mathcal{S}^2}$ est appelée *matrice de transition de la chaîne de Markov*.

Probabilité d'une trajectoire

Soit $(x_0, \ldots, x_n)$ une trajectoire. Sa vraisemblance est

$$\mathbb{P}(X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = \prod_{i=0}^{n-1} p_{x_i x_{i+1}} \pi_0(x_0) = \pi_0(x_0) \prod_{q,r} p_{qr}^{n_{qr}}$$

On peut alors estimer les probabilités de transition par $\hat{p_{qr}} = \frac{n_{ST}}{n}$.

Mesure invariante

Pour tout $n \geq 1$,

$$\pi_n^t = \pi_{n-1}^t P \tag{5}$$

Par conséquent,

$$\pi_n^t = \pi_0^t P^n \tag{6}$$

La deuxième égalité implique que la suite des distributions converge si et seulement la suite des puissances de la matrice de transition converge. La première implique que si une limite μ existe pour les distributions π_i, elle vérifie

$$\mu^t = \mu^t P$$

Une mesure vérifiant cette égalité est appelée mesure invariante de la chaîne de Markov.

Classification : chaîne périodique

Définition

Une chaîne est périodique si il existe $k \geq 2$ tel que sa matrice de transition vérifie $P^k = I$. Si ce n'est pas le cas, elle est dite apériodique.

Remarque : En pratique, on construit toujours les chaînes de façon à ce qu'elles soient non périodiques.

Classification : état récurrent/transient

Soit v un état et p la probabilité, étant donné que le point de départ de la chaîne est v, de revenir en v.

- Si $p = 1$, v est un état récurrent.
- En particulier, si $p_{vv} = 1$, l'état est dit absorbant.
- Si $p < 1$, l'état est dit transient ou transitoire.
  Dans ce cas, le nombre de passages en v de la marche est presque sûrement fini et suit une loi géométrique de paramètre p.

Classification : état récurrent/état transient

- Une chaîne de Markov peut etre représentée par un graphe orienté G : (u, v) est une arête ssi $p_{uv} \neq 0$.
- Une *composante fortement connexe* du graphe est un ensemble maximal S de sommets vérifiant la propriété suivante : pour toute paire de sommets u et v de S, il existe un chemin dirigé de u vers v et un chemin dirigé de v vers u.
- Soit H un graphe ayant un sommet pour chaque composante fortement connexe de G et tel que $(u, v) \in E(H)$ s'il existe une arête de G allant de la composante correspondant à u à la composante correspondant à v. Alors le graphe H est acyclique. De plus, les états récurrents sont les états situés dans les composantes connexes dont le degré sortant dans H est nul.

Definition

Une chaîne de Markov est *irréductible* si le graphe associé est fortement connexe, ou autrement dit s'il existe un chemin entre toute paire d'états.

Classification : Etat récurrent positif

- On considère une marche aléatoire symétrique sur $\mathbb{Z}$, pour laquelle à chapque étape on fait un pas vers la gauche ou vers la droite avec probabilité $\frac{1}{2}$. On peut démontrer que dans ce cas la probabilité de retour en 0 est de 1 mais que l'espérance du temps de retour en 0 est infinie.

Définition

Un état est récurrent positif s'il est récurrent et que l'espérance du temps de retour en cet état, partant de cet état, est fini. En d'autres termes, si la chaîne passe en fois par cet état, elle y passera infiniment souvent.

- Si la chaîne est finie et irréductible, tout état est récurrent positif ;
- On considère une marche aléatoire non symétrique sur $\mathbb{Z}$, telle qu'on fait un pas vers 0 avec probabilité $p > \frac{1}{2}$ et un pas opposé avec probabilité $q = 1 - p$. On peut alors montrer que 0 est récurrent positif.

Théorème de convergence

Théorème

On considère une chaîne de Markov irréductible et apériodique, admettant un état récurrent positif. Alors tous les états sont récurrent positifs et il existe une unique mesure invariante pi.
De plus, *quel que soit la mesure initiale* π_0, la suite des lois π_n des X_n converge vers π.

La démonstrationse fait en deux étapes

1. unicité de la mesure invariante : théorème de Perron-Frobenius
2. convergence vers la mesure invariante

Théorème de Perron-Frobenius

Théorème

Soit P la matrice d'une chaîne de Markov irréductible. Alors :

1. 1 est une valeur propre simple.
2. tout vecteur propre à gauche associé à 1 a toutes ses coordonnées de même signe. En particulier, celui de somme 1 correspond bien à une distribution de probabilités.
3. si la chaîne est apériodique, toute autre valeur propre λ vérifie $|\lambda| < 1$.

En d'autres termes, toute chaîne de Markov irréductible admet une unique mesure de probabilité invariante.

Convergence

Théorème

Soit P la matrice d'une chaîne de Markov irréductible et apériodique et μ l'unique mesure invariante associée. Alors, pour tout X_0, $\lim_{n\to+\infty}^{t} \pi_0 P^n =^t \mu$. De plus, la vitesse de convergence est en $|\lambda_2|^n$, où λ_2 est la valeur propre de valeur absolue maximale parmi les valeurs propres différentes de 1.

Idée de la démonstration : Ecrire ${}^t\pi_0$ dans une base de vecteurs propres à gauche de P.

Théorème ergodique

Théorème

On considère une chaîne de Markov irréductible apériodique de mesure invariante $\sum_{q \in \mathcal{S}} \pi_q |f(q)| < +\infty$.
Alors,

$$\lim_{n \to +\infty} \frac{1}{n} \sum_{i=0}^{n} f(X_i) = \sum_{q \in \mathcal{S}} \pi_q f(q)$$

Pour simuler un échantillon suivant la loi π, il n'est pas nécessaire de faire converger un grand nombre de chaînes : il suffit de mener une chaîne suffisamment longue.

Définition

Une suite de variables aléatoires $(X_i)_{i\geq 0}$ définies sur un ensemble $\mathcal{X}$ est une chaîne de Markov si $X_{i+1}|X_0, \ldots, X_i$ suit la même loi que $X_{i+1}|X_i$.

La fonction K telle que

$$X_{i+1}|X_0, \ldots, X_i \sim K(X_i, X_{i+1})$$

est appelé noyau markovien. Si f_i désigne la densité de X_i, on a alors

$$f_{i+1}(y) = \int_{\mathcal{X}} K(x, y) f_i(x)\, dx$$

Exemple : La marche aléatoire sur $\mathbb{R}$ définie par $X_{i+1} = X_i + \epsilon_i$ avec $\epsilon_i \sim \mathcal{N}(0, 1)$ est une chaîne de Markov dont le noyau $K(X_i, X_{i+1})$ correspond à la densité de $\mathcal{N}(X_i, 1)$.

Chaîne irréductible

Définition

La chaîne est irréductible si pour tout choix de la valeur initiale et tout ensemble A de mesurre non nulle, la probabilité que la chaîne atteigne A est non nulle.

Exemple :

- Marche aléatoire $X_{i+1} = X_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, 1)$.
- Marche aléatoire $X_{i+1} = X_i + \epsilon_i$, $\epsilon_i \sim \mathcal{U}[0, 1]$.
- Modèle AR(1) $X_{i+1} = aX_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, 1)$.

Loi limite

Si la chaîne est irréductible, il existe une unique loi stationnaire f qui est presque surement la loi limite de la chaîne de Markov.

Théorème ergodique

On considère une chaîne de Markov de distribution limite f. Pout toute fonction intégrable h,

$$\lim_{n \to +\infty} \frac{1}{n} \sum_{i=0}^{n-1} h(X_i) = \int_{\mathcal{X}} h(x) f(x) \, dx$$

En particulier, en prenant pour h la fonction indicatrice d'un sous-ensemble A de $\mathcal{X}$, on obtient que la mesure de A suivant f est égale à la proportion des éléments de la chaîne appartenant à A quand la chaîne devient infinie.
En d'autres termes, **générer une chaîne suffisamment longue de noyau K revient à simuler suivant f.**

III. Algorithmes de Metropolis-Hastings

III.1 Algorithme général

Principe

Points communs avec la méthode d'acceptation-rejet

- f une *distribution cible*, suivant laquelle on cherche à simuler.
- q une distribution de proposition, selon laquelle on va échantillonner, en acceptant la proposition avec une probabilité donnée

Différence

- q dépend de la valeur précédente de l'échantillon : $q() = q(.|x)$
- Si la proposition à partir de x_n est refusée, on pose $x_{n+1} = x_n$: l'échantillon contiendra des valeurs répétées
- L'échantillon correspond à une trajectoire d'une chaîne de Markov.

Algorithme de Metropolis-Hastings

Etant donné x_n,

1. Générer $y_n \sim q(y|x_n)$
2. Choisir

$$x_{n+1} = \begin{cases} y_n & \text{avec probabilité} \quad \rho(x_n, y_n) \\ x_n & \text{avec probabilité} \quad 1 - \rho(x_n, y_n) \end{cases}$$

où

$$\rho(x, y) = \min \left\{ \frac{f(y)}{f(x)} \frac{q(x|y)}{q(y|x)}, 1 \right\}$$

Algorithme de Metropolis-Hastings

Théorème

Les (x_n) forment une chaîne de Markov. Si q est tel que cette chaîne est irréductible, sa distribution limite est f.

- Toute loi de proposition q rendant la chaîne irréductible convient
- Contrairement à l'algorithme d'acceptation-rejet, on a plus besoin d'évaluer $\max \frac{f}{q}$.

Le choix de q n'influe pas sur le fait qu'il y a convergence, mais il influe sur la vitesse de celle-ci. Certains choix sont privilégiés.

III.2 Algorithme indépendant

Algorithme de Metropolis-Hastings indépendant

- la proposition ne dépend pas de la valeur courante.

Etant donné x_n,

1. Générer $y_n \sim g(y)$
2. Choisir

$$x_{n+1} = \begin{cases} y_n & \text{avec probabilité} \quad \rho(x_n, y_n) \\ x_n & \text{avec probabilité} \quad 1 - \rho(x_n, y_n) \end{cases}$$

où

$$\rho(x, y) = \min \left\{ \frac{f(y)}{f(x)} \frac{g(x)}{g(y)}, 1 \right\}$$

Algorithme de Metropolis-Hastings indépendant

- très ressemblant à l'acceptation-rejet
- pas besoin d'évaluer $\max \frac{f}{g}$, mais on perd l'indépendance entre les valeurs de l'échantillon
- la convergence sera d'autant plus rapide que la distribution g est proche de f

Exemple

- jeu de données `esoph` sous `R` : nombre de cancer de l'oesophage et de patients sains dans un échantillon stratifié suivant l'âge, la consommation d'alcool et la consommation de tabac.
- Y_i la variable aléatoire correspondant à l'indicatrice du fait que l'individu i développe un cancer de l'oesophage.
- modèle de régression logistique :

$$\log\left(\frac{\mathbb{P}(Y_i = 1)}{1 - \mathbb{P}(Y_i = 1)}\right) = \alpha + \beta Age_i + \gamma Tab_i + \delta Alc_i$$

Question

Trouver un intervalle de confiance de niveau 95% pour la probabilité de développer un cancer pour un individu dont les variables Age_i, Tab_i et Alc_i sont connues.

Exemple

- $\boldsymbol{\theta} = (\alpha, \beta, \gamma, \delta)$ le vecteur des paramètres, $\mathbf{X}_i = (1, Age_i, Tab_i, Alc_i)$ le vecteur des données de l'individu i. La vraisemblance est

$$\log \mathcal{L}(\boldsymbol{\theta}) = \sum_i \log \frac{exp({}^t\boldsymbol{\theta}\, X_i)}{1 + exp({}^t\boldsymbol{\theta}\, X_i)}$$

- On se place un cadre bayésien, avec une loi à priori pour laquelle les quatre coefficients sont indépendants, de loi normale centrée réduite
- ϕ la densité de la gaussienne centrée réduite

$$\mathbb{P}(\boldsymbol{\theta} \mid \mathbf{X}) \propto \prod_i \frac{exp({}^t\boldsymbol{\theta}\, X_i)}{1 + exp({}^t\boldsymbol{\theta}\, X_i)} \times \prod_{k=1}^{4} \log \phi(\theta_k)$$

On cherche à simuler suivant cette dernière distribution pour obtenir des intervalles de confiance.

Exemple

- L'estimateur du maximum de vraisemblance $\hat{\theta}$ peut être déterminé.

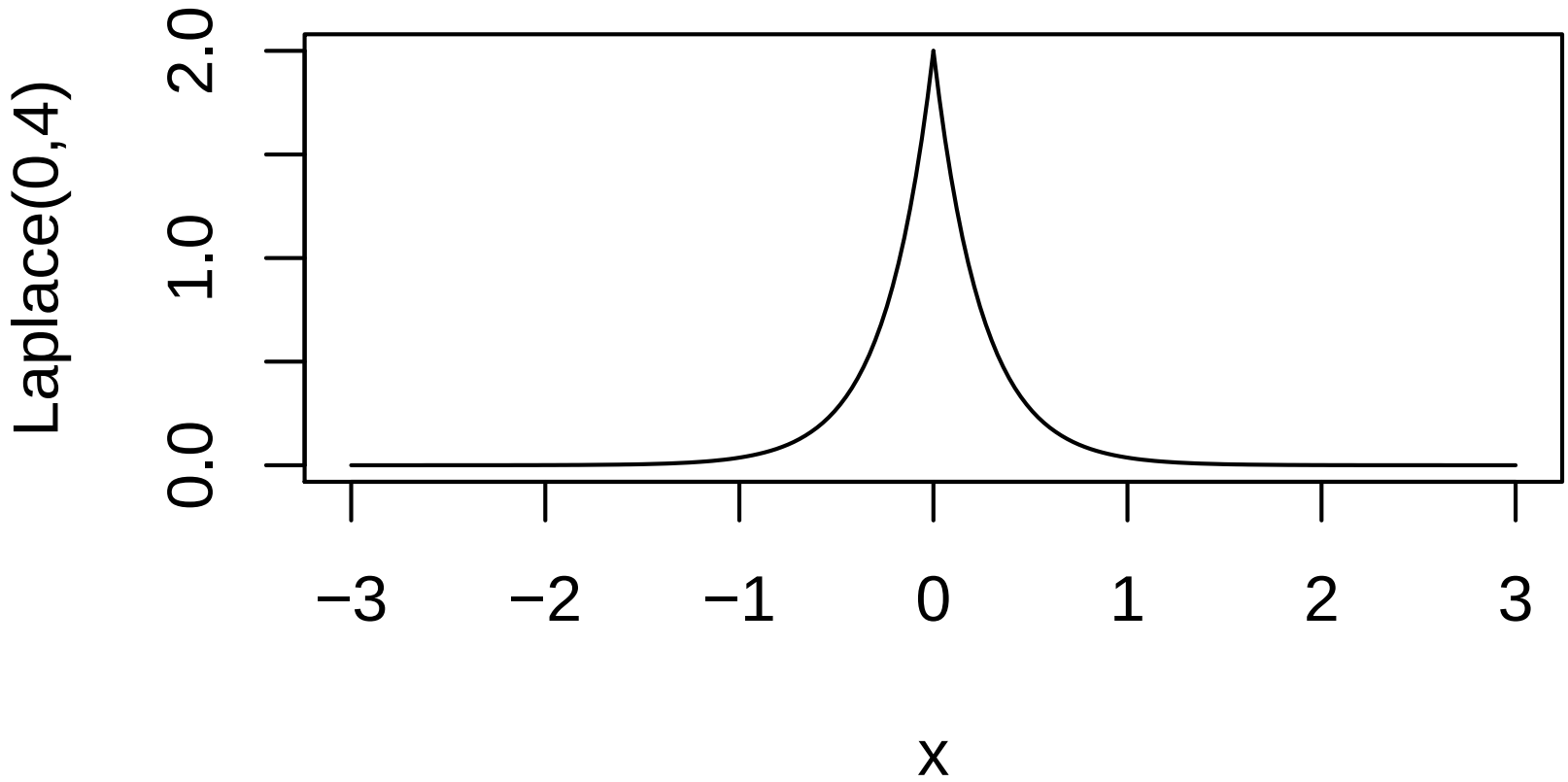
```
> model <- glm(cbind(ncases,ncontrols) ~ unclass(agegp)+unclass(alcgp)+
> EMV <-  model$coefficients
> EMV
```

```
  (Intercept) unclass(agegp) unclass(alcgp) unclass(tobgp)
   -5.5959444      0.5286674      0.6938248      0.2744565
```

Exemple

- La loi de Laplace de paramètres (μ, b) est la loi continue de densité

$$\forall x \in \mathbb{R}, f(x) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$



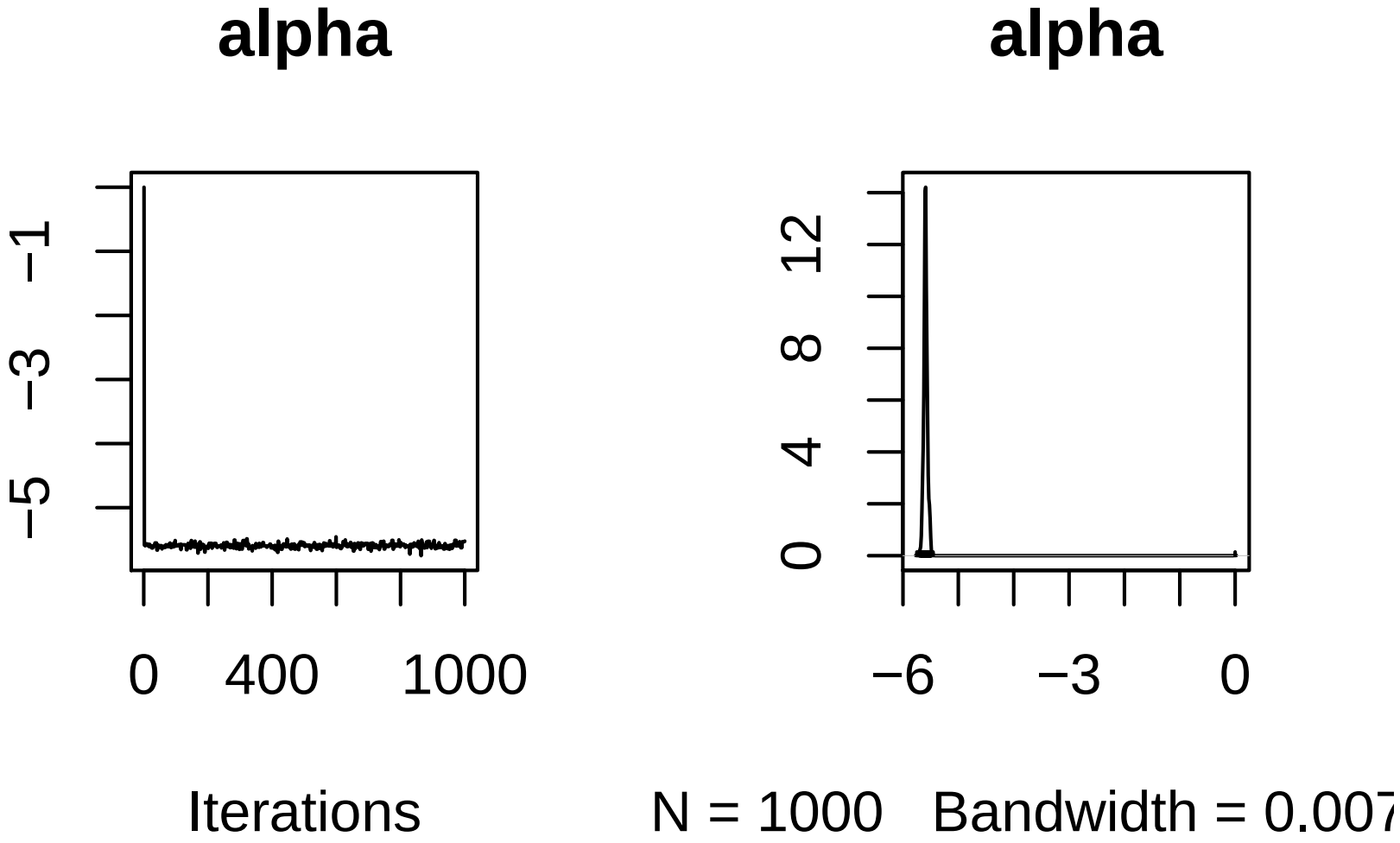
Exemple

```
> #Calcul de la vraisemblance à une constante près pour une valeur de Theta
> logit <- function(x){
+   return(exp(x)/(1+exp(x)))
+ }
> LogLikelihood <- function(Theta, data){
+   logL <- 0
+   coeffmatrix <- cbind(1,data$agegp,data$alcgp,data$tobgp) #matrice des coefficients correspondant à chaque possibilité
+   for (i in 1:dim(data)[1]){
+     proba <- logit(t(Theta)%*%coeffmatrix[i,])
+     logL <- logL+log(proba)*data$ncases[i]+log(1-proba)*data$ncontrols[i]
+   }
+   logL <- logL + sum(log(dnorm(Theta)))   # ajouter la loi à priori où chacune prise comme loi normale central réduite
+   return(logL)
+ }
> #MCMC par Metropolis-Hastings indépendant.
> rlaplace <- function(n,mu,b){
+   sign <- -1 + 2*floor(runif(n,0,2))
+   return(mu + sign*rexp(n,b))
+ }
> dloglaplace <- function(x,mu,b){
+   return(-log(2*b)-abs(x-mu)/b)
+ }
>
```

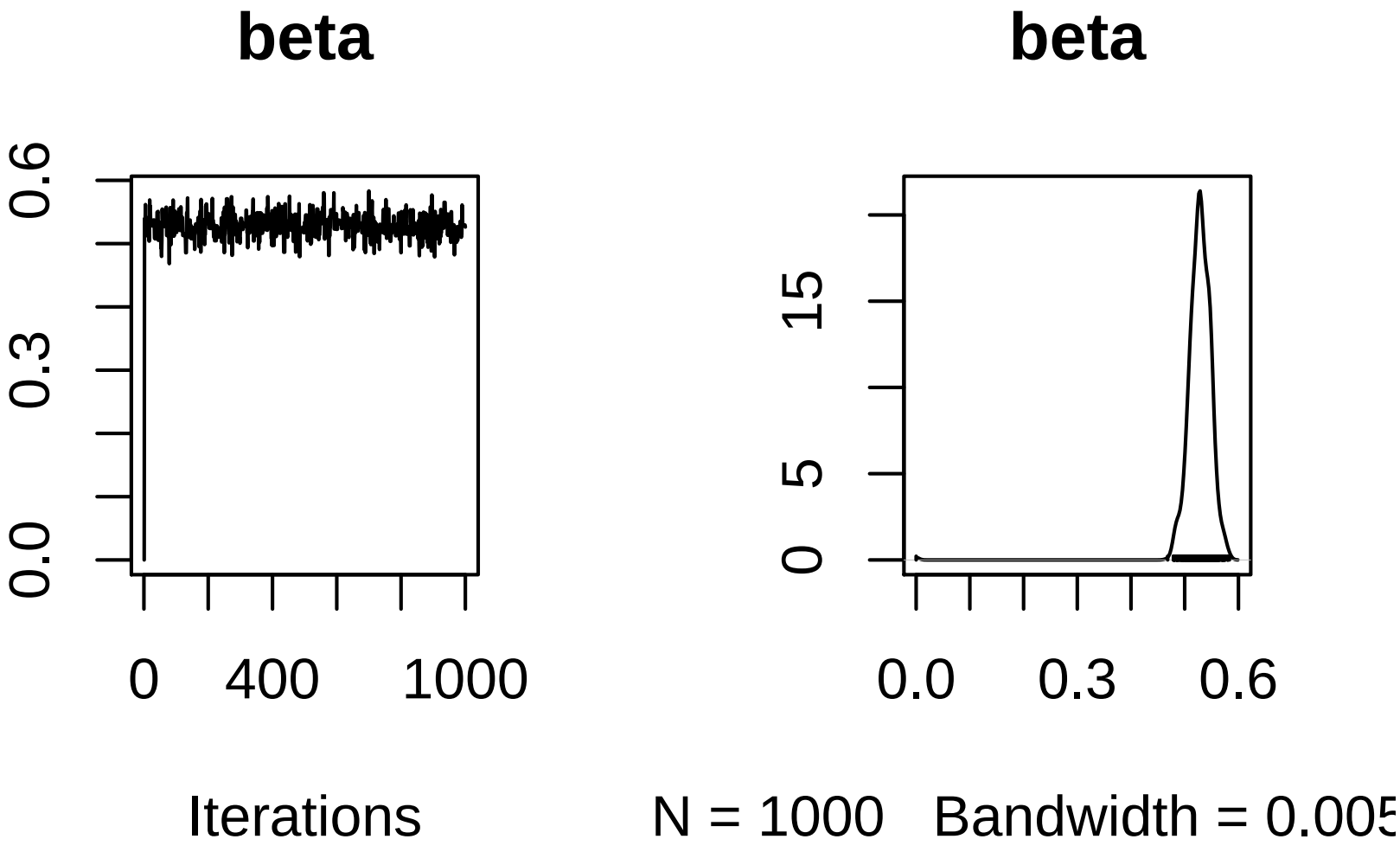
Exemple

```
> propositionInd <- function(EMV,B){  #proposition indépendante suivant des lois de Laplace centrées sur les EMV et dont les aut
+   return(c(rlaplace(1,EMV[1],B[1]),rlaplace(1,EMV[2],B[2]),rlaplace(1,EMV[3],B[3]),rlaplace(1,EMV[4],B[4])))
+ }
> trajectoryInd <- function(Nsim,EMV,B,data,X0){       #generation d'une trajectoire de longueur Nsim
+
+   X <- matrix(X0,1,4)
+   proba <- 0
+   for (n in 2:Nsim){
+     Y <- propositionInd(EMV,B)
+     logrho <- LogLikelihood(Y,data)-LogLikelihood(X[n-1,],data)-dloglaplace(Y[1],EMV[1],B[1])+dloglaplace(X[n-1,1],EMV[1],B[1]
+     accept <- (runif(1)<exp(logrho))
+     X <- rbind(X,X[n-1,]*(1-accept) + Y * accept)
+     s <- t(X[n,])%*%c(1,1,3,1)
+     proba <- c(proba,exp(s)/(1+exp(s)))
+   }
+   return(list(X=X,proba=proba))
+ }
> data <- esoph
> data$tobgp <- unclass(data$tobgp)
> data$alcgp <- unclass(data$alcgp)
> data$agegp <- unclass(data$agegp)
> trajectory1 <- trajectoryInd(1000,EMV,c(40,40,40,40),data,c(0,0,0,0))
> xInd <- as.mcmc(trajectory1$X)
> prInd <- as.mcmc(trajectory1$proba)
>
```

Exemple


alpha
−5
−3
−1
0
400
1000
Iterations
alpha
0
4
8
12
−6
−3
0
N = 1000 Bandwidth = 0.007

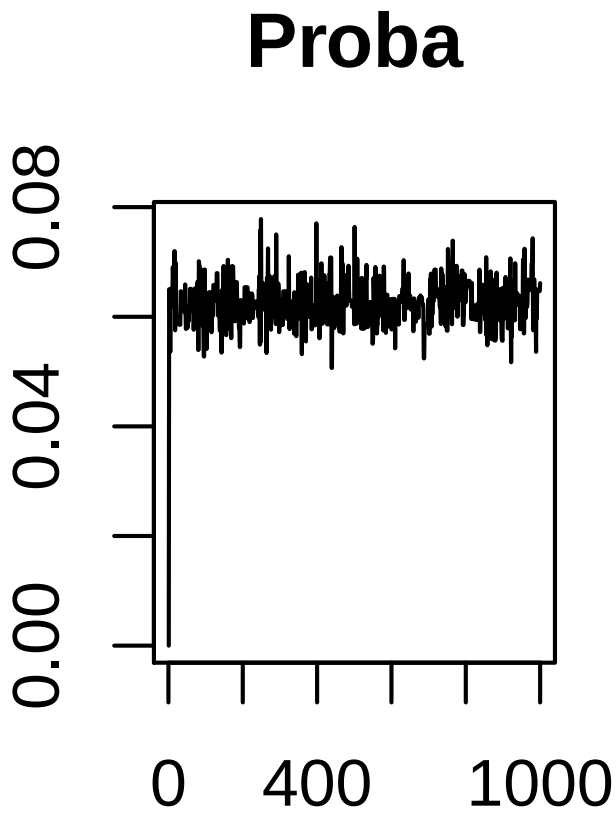
Exemple

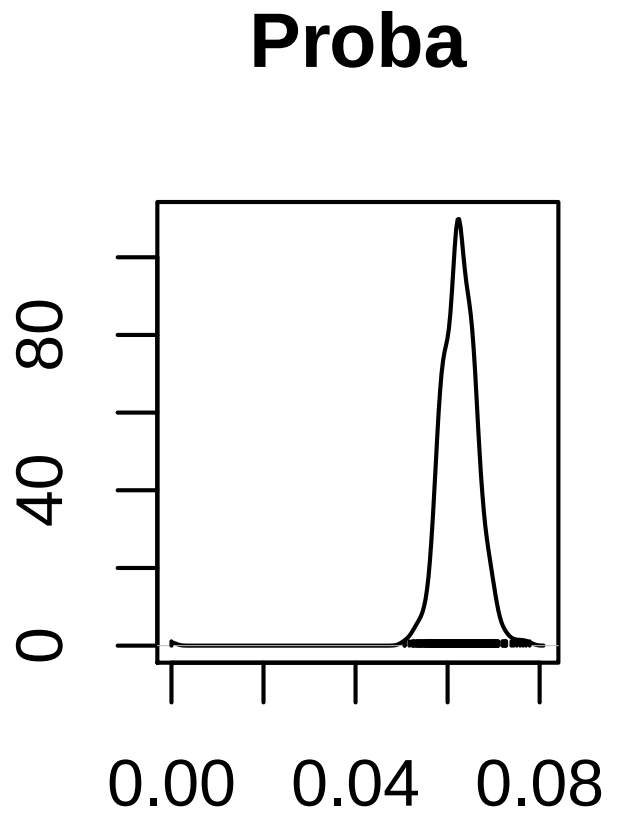

beta
0.0
0.3
0.6
0
400
1000
Iterations
beta
0
5
15
0.0
0.3
0.6
N = 1000 Bandwidth = 0.005

Exemple

On considère un non-fumeur ($F_i = 1$), consommateur moyen d'alcool ($A_i = 3$) de 30 ans. On peut déterminer à chaque étape de la simulation la probabilité de développer un cancer de l'oesophage.
On récupère alors une simulation de la distribution de cette probabilité

```
> plot(prInd,main='Proba')
```





III.3 Convergence

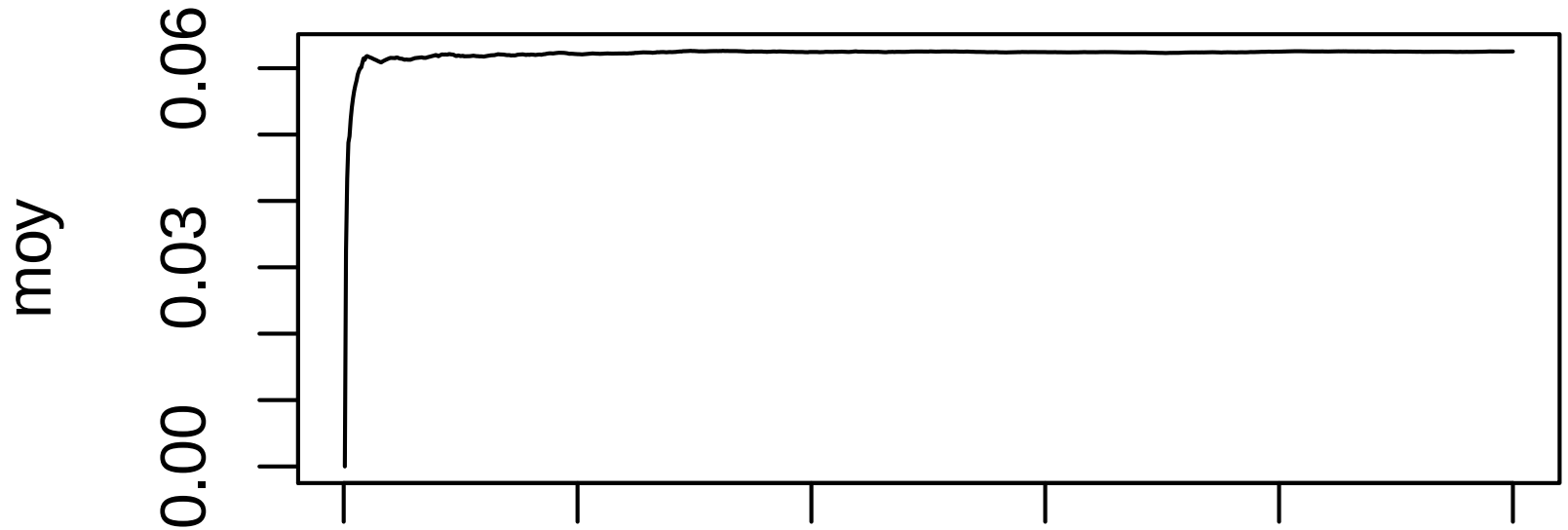
Masse manquante

- tout algorithme MCMC converge vers la bonne distribution
- on ne sait jamais si on a déjà convergé ou pas : il peut rester une partie de l'espace où la distribution n'est pas nulle mais qui n'a pas encore été exploré. On parle de masse manquante
- il faut toujours faire tourner de tels algorithmes le plus longtemps possible !

- il y a certains critères que l'on peut tout de même vérifier

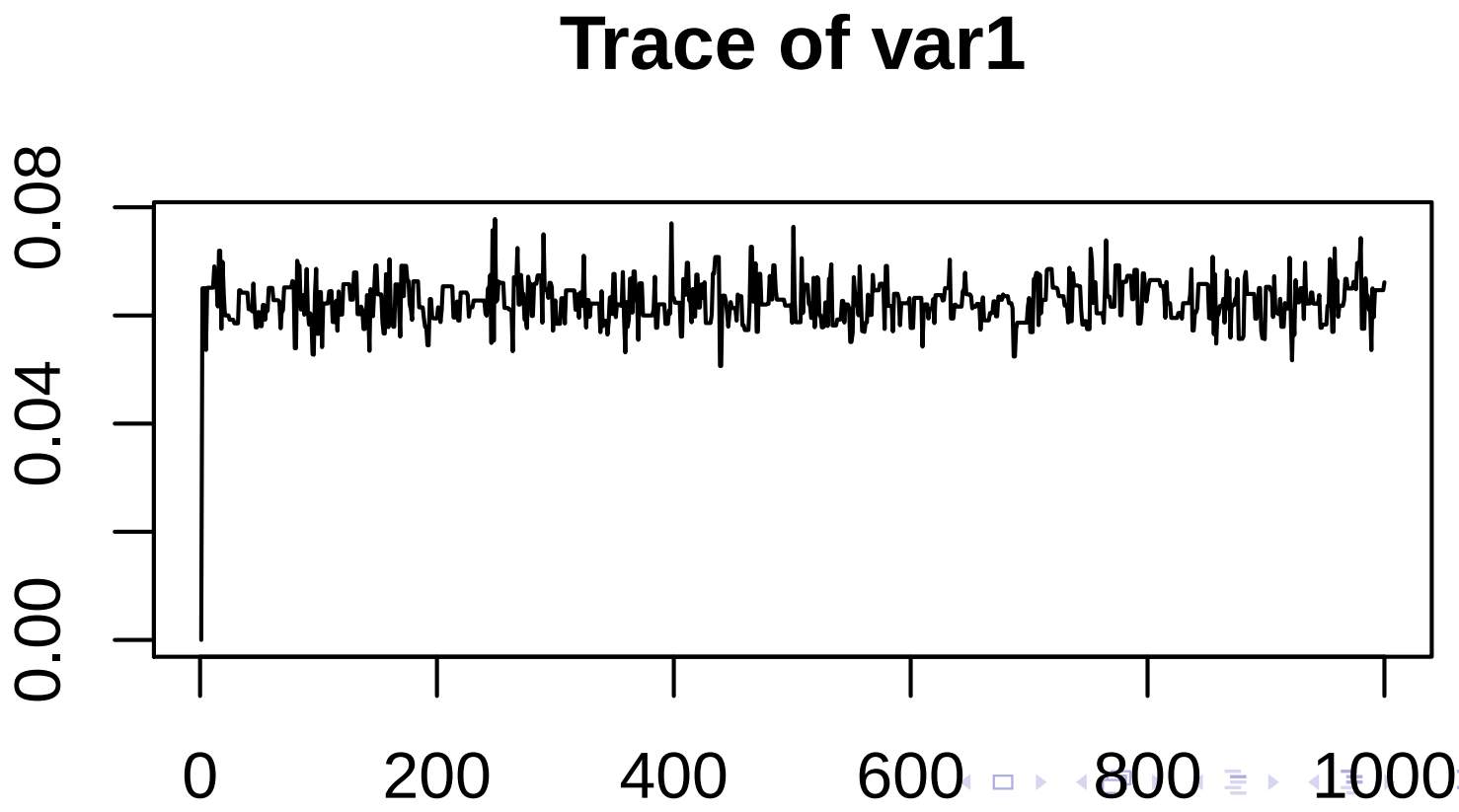
Convergence de la moyenne

- Le théorème ergodique entraîne que lorsque la chaîne a atteint sa distribution limite, la moyenne de $\frac{1}{n}\sum_{i=1}^{n} x_i$ tend vers $\mathbb{E}(f)$ quand n tend vers l'infini. Tant que cette courbe ne se stabilise pas, on est donc sûr de ne pas avoir convergé.
- Cela dit, la convergence de cette quantité ne veut pas dire que tout l'espace est bien parcouru. Par exemple, si la distribution est multimodale est que la chaîne est enfermée dans un mode, la courbe converge alors que l'échantillonnage suivant f est mauvais.



Evaluation du *mixing*

- Si on trace la courbe des x_i, on doit voir apparaître à convergence une courbe qui oscille dans tout le domaine des valeurs possibles.
- Cela se voit relativement bien à l'oeil nu quand en dimension 1. En dimension plus grande, on peut parfois repérer une comportement anormal.
- Lancer parallèlement plusieurs chaînes peut être une manière de repérer un problème de mixing si les régions parcourues par les différentes chaînes ne sont pas les mêmes.



Probabilité d'acceptation

- L'espérance de la probabilité d'acceptation

$$\overline{\rho} = \int \rho(x, y) f(x) q(y|x) dy dx$$

  est appelé *taux d'acceptation*.
- Comme la loi de la chaîne des (x_n) tend vers $f(x)$, la loi du couple (x_n, y_n) tend vers $f(x)q(y|x)$, et le théorème ergodique des chaînes de Markov entraîne

$$\overline{\rho} = \lim_{n \to +\infty} \frac{1}{n+1} \sum_{i=0}^{n} \rho(x_n, y_n)$$

- Une probabilité d'acceptation trop faible indique qu'on est trop souvent sur des valeurs extrêmes de f, l'espace entre les maxima n'est donc pas bien exploré. Une probabilité d'acceptation trop forte signifie qu'on accepte presque systématiquement le mouvement, ce qui prouve qu'on ne passe pas assez de temps près des maxima de f.
- Une règle basée sur des constatations empiriques recommande un taux d'acceptation de l'ordre de $\frac{1}{2}$ pour les modèles en petite dimension (1 ou 2) et un taux de l'ordre de $\frac{1}{4}$ en dimension plus grande.

```
> rho <- 1 - sum(c(0,prInd)==c(prInd,0))/length(prInd)
> rho
[1] 0.497
```

Autocorrélation

- Considérons une chaîne de Markov évoluant sous sa loi stationnaire f. La fonction d'autocovariance $\gamma : \mathbb{N} \to \mathbb{R}$ est définie par

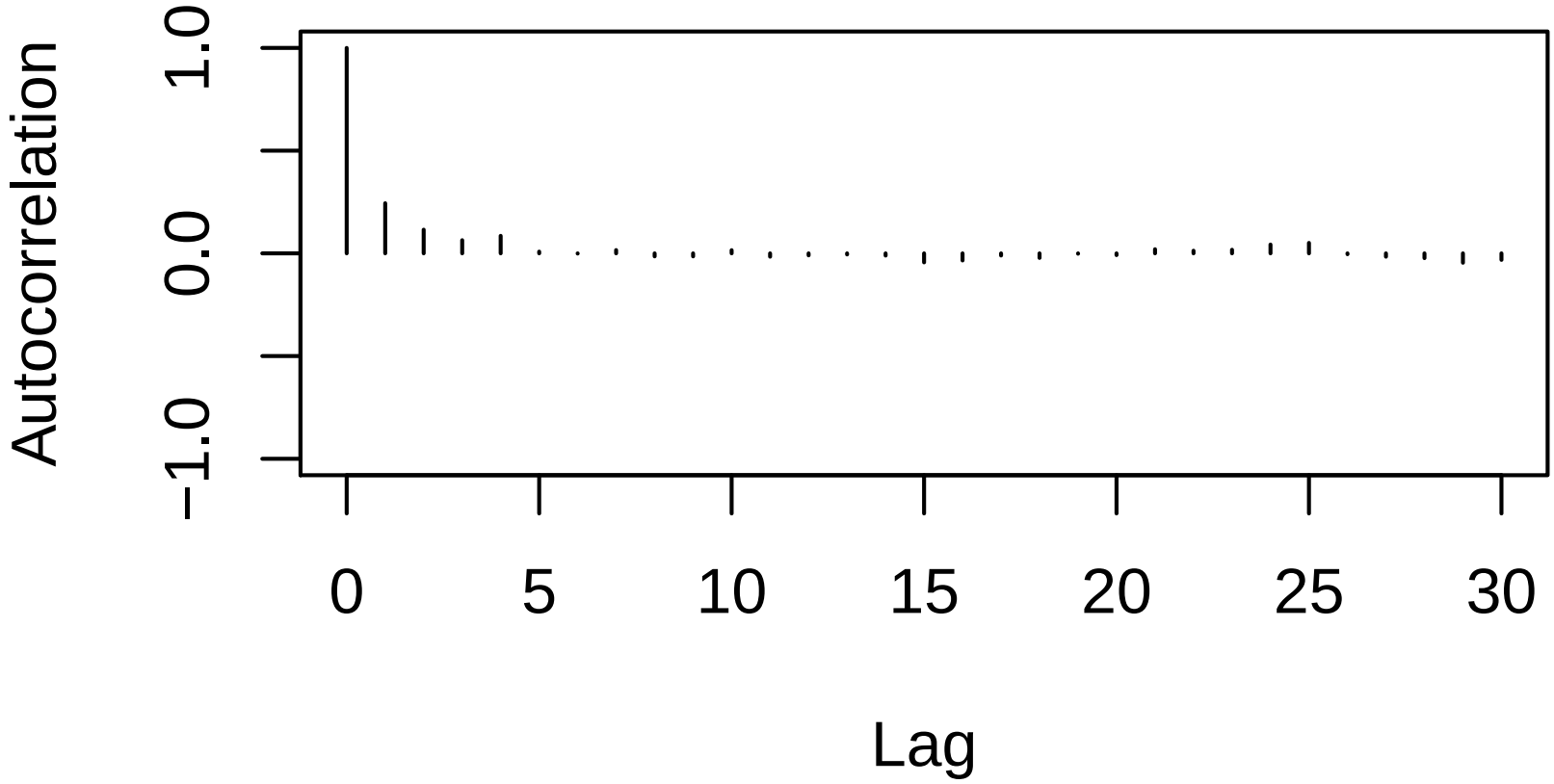
$$\gamma(k) = cov(f(X_i), f(X_{i+k}))$$

L'autocorrélation est obtenue en divisant l'autocovariance par γ_0.
Pour une trajectoire assez longue de la chaîne, l'autocovariance peut être estimée par

$$\hat{\gamma}(k) = \frac{1}{n} \sum_{i=1}^{n-k} [f(X_i) - \hat{\mu}][f(X_{i+k} - \hat{\mu}]$$

- Cet outil permet de voir à quel point la chaîne oublie l'influence du passé. Une autocovariance diminuant trop lentement est mauvais signe car elle indique que la chaîne ne parcourt pas assez vite des zones nouvelles du domaine. On estime qu'il faut mener des chaînes de longueur égales à un grand nombre de fois la valeur de k nécessaire pour que l'autocorrélation devienne proche de 0.

Autocorrélation



III.4 Algorithme par marche aléatoire

Algorithme de Metropolis-Hastings par marche aléatoire

- Les propositions suivent une marche aléatoire symétrique

1. Générer $y_n \sim g(y - x_n)$, g symétrique
2. Choisir

$$x_{n+1} = \begin{cases} y_n & \text{avec probabilité} \quad \rho(x_n, y_n) \\ x_n & \text{avec probabilité} \quad 1 - \rho(x_n, y_n) \end{cases}$$

où

$$\rho(x, y) = \min \left\{ \frac{f(y)}{f(x)}, 1 \right\}$$

Algorithme de Metropolis-Hastings par marche aléatoire

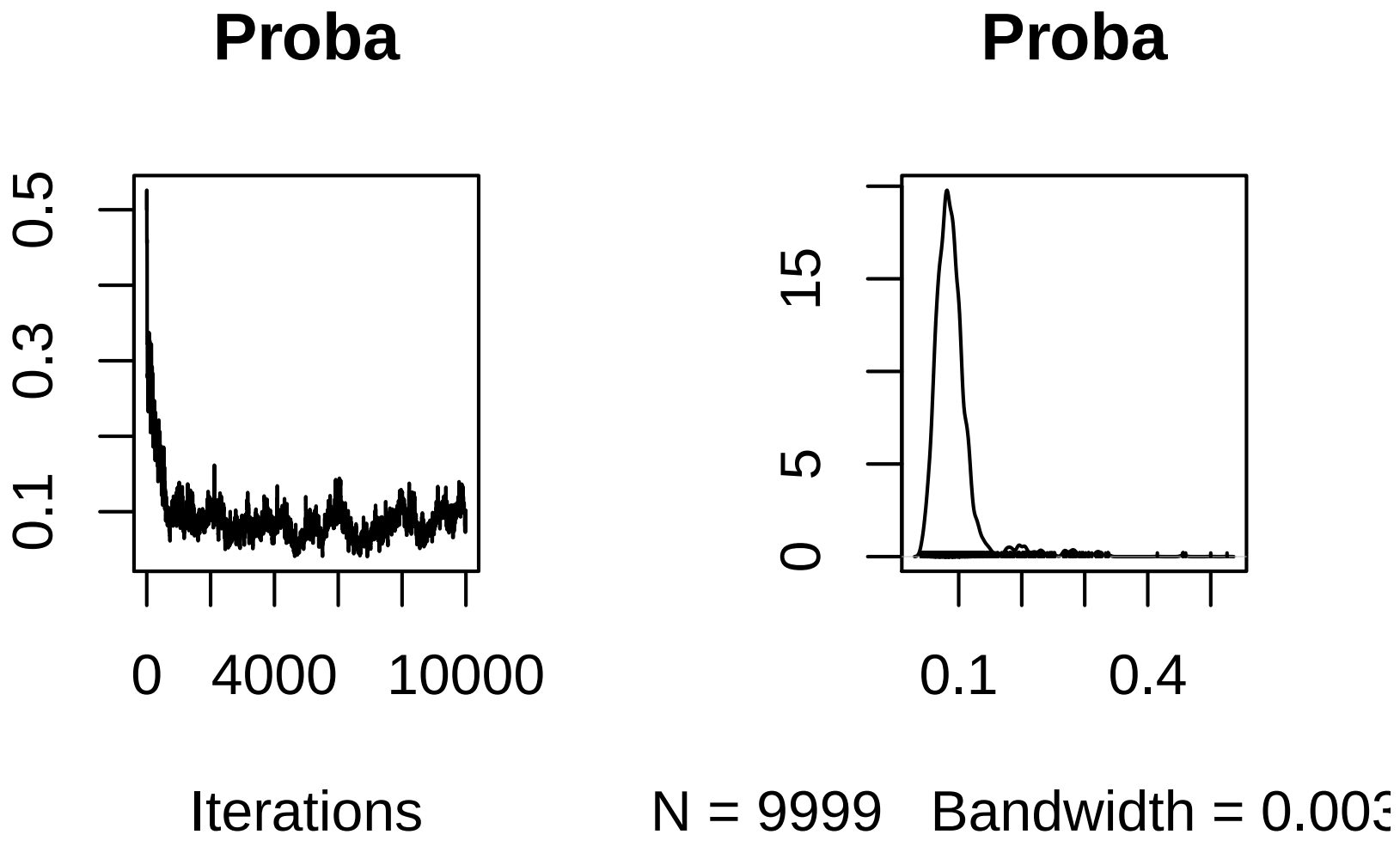
- la probabilité d'acceptation ne dépend plus de g. La chaîne en dépend via les propositions.
- approche très simple qui peut être utilisée quand f est très mal connue, contrairement à l'algorithme indépendant : espaces de grande dimension et/ou espaces d'objets discrets.
- il faut que la marche parcoure l'espace de tous les possibles.

Exemple

```
> trajectoryRW <- function(Nsim,data,width,X0){
+
+   X <- matrix(X0,1,4)
+   proba <- c()
+   for (n in 2:Nsim){
+     Y <- runif(4,-width,width)
+     rho <- exp(LogLikelihood(X[n-1,]+Y,data) - LogLikelihood(X[n-1,],data))
+     X <- rbind(X, X[n-1,] + Y * (runif(1)<rho))
+     if (floor(n/100)==(n/100)) { print(n)}
+     s <- t(X[n,])%*%c(1,1,3,1)
+     proba <- c(proba,exp(s)/(1+exp(s)))
+   }
+   return(list(X=X,proba=proba))
+ }
> data <- esoph
> data$tobgp <- unclass(data$tobgp)
> data$alcgp <- unclass(data$alcgp)
> data$agegp <- unclass(data$agegp)
> trajectory <- trajectoryRW(10000,data,.1,c(0,0,0,0))
[1] 100
[1] 200
[1] 300
[1] 400
[1] 500
[1] 600
[1] 700
[1] 800
[1] 900
[1] 1000
[1] 1100
[1] 1200
[1] 1300
[1] 1400
[1] 1500
[1] 1600
[1] 1700
[1] 1800
[1] 1900
[1] 2000
[1] 2100
[1] 2200
[1] 2300
```

Exemple

```
> plot(prRW,main='Proba')
```



Avantages

- l'influence du choix de la distribution de proposition est plus faible que dans le cas indépendant
- cela permet de gérer des espaces de dimensions très grandes, dans lesquels simuler intelligemment une proposition indépendante est difficile.
  Exemple : Parcourir l'ensemble des arbres phylogéniques ou l'ensemble des variables explicatives à insérer dans un modèle linéaire.

Calibration de la proposition

- des propositions trop proches du point courant vont favoriser des marches qui restent toujours dans la même région : risque d'avoir raté des pans entiers de l'espace à explorer au moment où on arrête la simulation.
  De plus, si la distribution est multimodale, la marche risque de rester enfermée dans un mode car il faudrait accepter successivement un grand nombre de sauts défavorables pour en sortir (ce qui arrive avec probabilité non nulle mais tellement faible qu'on ne le voit jamais).
- des propositions à trop longue distance ne sont pas forcément faciles à formuler. De plus, lorsqu'on est proche d'un maximum local de f, on risque d'y rester très longtemps avant d'accepter un mouvement, ce qui entraîne une chaîne très fortement corrélée.

III.5 Recuit simulé

Mesure de Gibbs

- On considère une fonction f que l'on souhaite minimiser
- La mesure de Gibbs associée à f et à la température T est définie par

$$\mu_T(x) = \frac{1}{Z_T} e^{-f(x)/T} \text{ avec } Z_T = \int_x e^{-f(x)/T} dx$$

- La mesure μ_T est maximale clairement en les minima de f. Cependant, si T est très faible, elle est beaucoup plus piquée que f. En effet,

$$\frac{\mu_T(x)}{\mu_T(x^*)} = exp(\frac{f(x^*) - f(x)}{T})$$

- A la limite, $\lim_{T \to 0} \mu_T(x) = 0$ si x n'est pas un minimum de f et $\lim_{T \to 0} \mu_T(x) = 1/k$ si f admet k minimum et que x est l'un d'eux.

Mesure de Gibbs

Plutôt que de simuler suivant f afin de trouver son minimum, il est par conséquent intéressant de simuler suivant μ_T avec un T petit. En effet, les cuvettes de f correspondant aux minima globaux ont alors une plus grande probabilité d'apparition. La difficulté apparente est le calcul de Z_T mais on peut s'en passer en simulant suivant un algorithme de Metropolis-Hastings.

Etant donné x_n,

1. Générer $\xi_n \sim g(\xi)$, g symétrique
2. Choisir

$$x_{n+1} = \begin{cases} x_n + \xi_n & \text{avec probabilité} \quad \rho(x_n, x_n + \xi_n) \\ x_n & \text{avec probabilité} \quad 1 - \rho(x_n, y_n) \end{cases}$$

où

$$\rho(x, y) = \min \left\{ \exp\left(\frac{f(x_n + \xi_n) - f(x_n)}{T}\right), 1 \right\}$$

Recuit simulé

- On reprend l'idée précédente en cherchant à simuler non pas toute une suite suivant μ_T, mais une suite (x_n) telle que (x_n) soit distribuée suivant μ_{T_n}, avec T_n tendant vers 0. On s'attend en effet à ce que dans ce cas, x_n tende vers un minimum global.

Etant donné x_n,

1. Générer $\xi_n \sim g(\xi)$, g symétrique
2. Choisir

$$x_{n+1} = \begin{cases} x_n + \xi_n & \text{avec probabilité} \quad \rho(x_n, x_n + \xi_n) \\ x_n & \text{avec probabilité} \quad 1 - \rho(x_n, y_n) \end{cases}$$

où

$$\rho(x, y) = \min\left\{ \exp\left(\frac{f(x_n + \xi_n) - f(x_n)}{T_n}\right), 1\right\}$$

Recuit simulé

- Algorithme très semblable à celui de Metropolis-Hastings par marche aléatoire : si le mouvement proposé représente un gain, il est systématiquement accepté, s'il représente une perte, il est accepté avec un probabilité d'autant plus petite que la perte est importante.
- La probabilité d'acceptation pour un perte donnée diminue avec l'allongement de la chaîne. En d'autres termes, la chaîne va accepter avec assez grande probabilité des mouvements non croissants au début de son mouvement, et les acceptera de plus en plus difficilement par la suite.

Théorème

Pour toute fonction f, il existe une constante C_f telle que $T_n \leq \frac{C_f}{\log n}$ entraîne que x_n tend vers un minimum global de f avec probabilité 1.

- Si la chaîne finit théoriquement toujours par converger, on ne sait pas quand elle l'a effectivement fait. Elle peut passer un temps très long dans un minimum local avant de finalement découvrir une nouvelle région plus intéressante.
- Un choix de forme logarithmique $T_n = \frac{C}{\log(n)}$ converge lentement mais a de meilleures chances de ne pas rester enfermée dans un minimum local, alors qu'un choix géométrique de la forme $T_n = \alpha^n T$, α proche de 1, donne plus rapidement une impression de convergence.

Exemple : le voyageur de commerce

Problème

- Un voyageur doit passer par n villes numérotées de 1 à n et revenir à son point de départ, la distance entre les villes étant donnée par la fonction d.
- Quelle est le meilleur choix d'itinéraire, c'est-à-dire la permutation σ minimisant $f(\sigma) = \sum_{i=1}^{n-1} d(\sigma(i), \sigma(i+1))$?

- Le problème est NP-complet : une heuristique est nécessaire.
- L'approche MCMC donne de bons résultats.

Exemple : le voyageur de commerce

```
> M <- matrix(runif(10000,.5,1.5),100,100)
> M <- M+t(M)                    #M est une matrice symétrique dont tout coe
> diag(M) <- 0
> for (i in 1:99){
+   M[i,i+1] <- 1
+   M[i+1,i] <- 1
+ }
> M[100,1] <- 1
> M[1,100] <- 1
```

Exemple : le voyageur de commerce

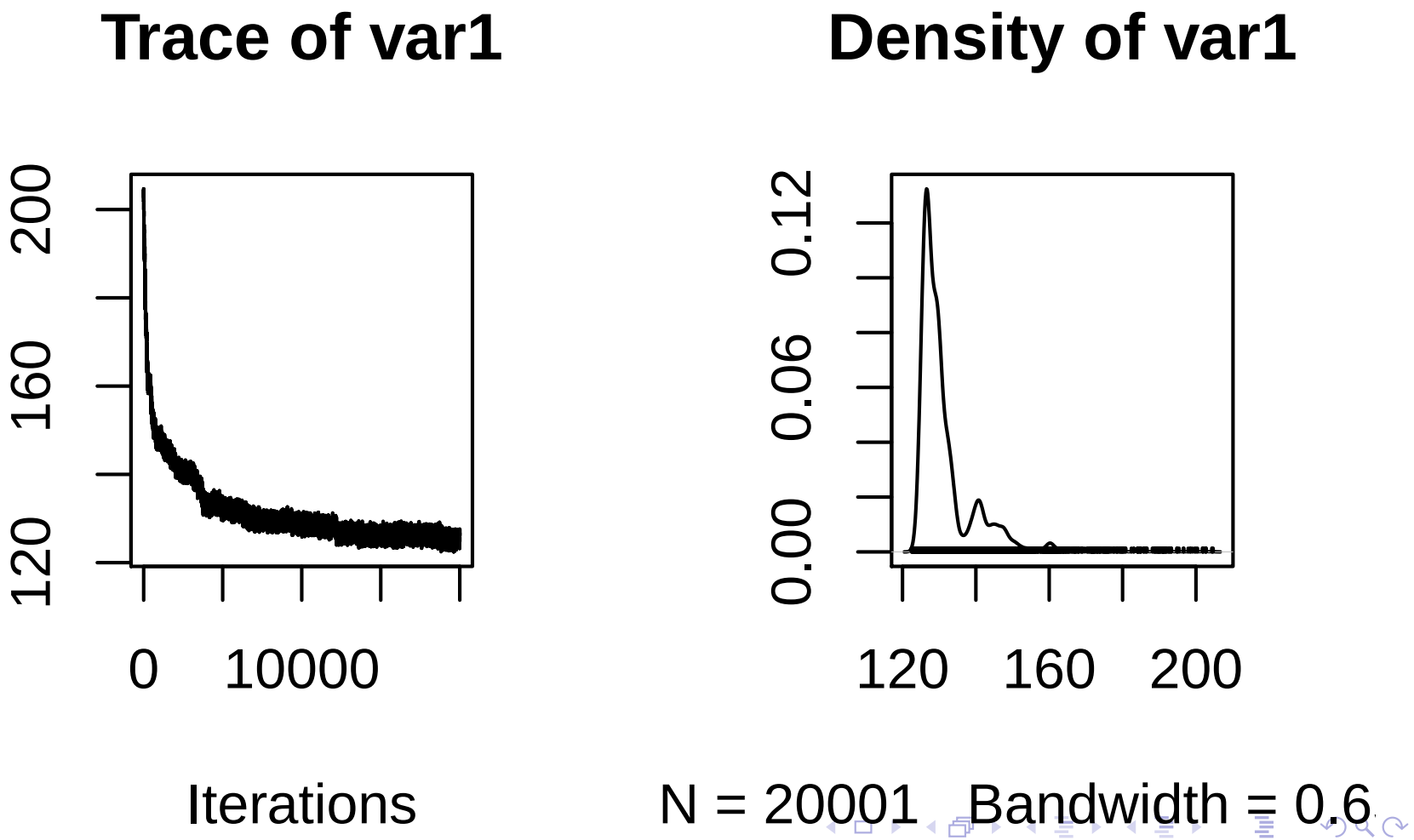
```
> cost <- function(M,sigma){ #cout du chemin correspondant au chemin si
+   cost <- M[sigma[100],sigma[1]]
+   for (i in 1:99){
+     cost <- cost + M[sigma[i],sigma[i+1]]
+   }
+   return(cost)
+ }
> shuffle <- function(sigma){
+   newsigma <- sigma
+   exchange <- sample(c(1:100),2,replace=FALSE)
+   newsigma[exchange[1]] <- sigma[exchange[2]]
+   newsigma[exchange[2]] <- sigma[exchange[1]]
+   return(newsigma)
+ }
```

Exemple : le voyageur de commerce

```
> simulatedAnnealing <- function(M,Temp){ #recuit simulé pour la matric
+
+   sigma <- sample(c(1:100),100,replace=FALSE)
+   cost <- cost(M,sigma)
+   costvector <- c(cost)
+
+   for (n in 1:length(Temp)){
+      newsigma <- shuffle(sigma)
+      newcost <- cost(M,newsigma)
+      costvector <- c(costvector,newcost)
+
+      rho <- exp((-cost(M,newsigma) + cost(M,sigma))/Temp[n])  #car on
+
+      if (runif(1)<rho){
+        sigma <- newsigma
+        cost <- newcost
+      }
+   }
+
+   return(costvector)
+ }
```

Exemple : le voyageur de commerce

```
> Temp <- 1 / log(1:20000)
> traj <- simulatedAnnealing(M,Temp)
> codatraj <- as.mcmc(traj)
> plot(codatraj)
```



IV. ECHANTILLONNAGE DE GIBBS

IV.1. Principe

Problèmes et notations

- On considère un problème multi-dimensionnel dans lequel on cherche à simuler la distribution d'un paramètre $\Theta = (\theta_1, \ldots, \theta_n)$ étant donné des observations.
- On suppose qu'on ne sait pas simuler la loi jointe $\mathbb{P}(\Theta|X)$
- On note Θ_{-i} le vecteur $\Theta = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n)$
- On suppose qu'on est capable de simuler, pour tout i, la loi $\mathbb{P}(\theta_i|X, \Theta_{-i})$

Principe

On considère le vecteur $\Theta^{(t)}$ obtenu à la t^{eme} iération. On en déduit le vecteur $\Theta^{(t+1)}$ comme suit

- on simule
$$\theta_1^{(t+1)} \sim \mathbb{P}(\theta_1 | X, \theta_2 = \theta_2^{(t)}, \ldots, \theta_n = \theta_n^{(t)})$$
- on simule
$$\theta_2^{(t+1)} \sim \mathbb{P}(\theta_2 | X, \theta_1 = \theta_1^{(t+1)}, \theta_3 = \theta_3^{(t)}, \ldots, \theta_n = \theta_n^{(t)})$$
- ...
- on simule
$$\theta_n^{(t+1)} \sim \mathbb{P}(\theta_2 | X, \theta_1 = \theta_1^{(t+1)}, \ldots, \theta_{n-1} = \theta_{n-1}^{(t+1)})$$

Convergence

- La suite des vecteurs générés est un chaîne de Markov irréductible.
- Supposons que $\Theta^{(t)} \sim \mathbb{P}(\Theta|X)$. Alors

$$(\theta_1^{(t+1)}, \theta_2^{(t)}, \ldots, \theta_n^{(t)}) \sim \mathbb{P}(\theta_1|\theta_2, \ldots, \theta_n, X)\mathbb{P}(\theta_2, \ldots, \theta_n|X)$$
$$\sim\sim \mathbb{P}(\Theta|X)$$

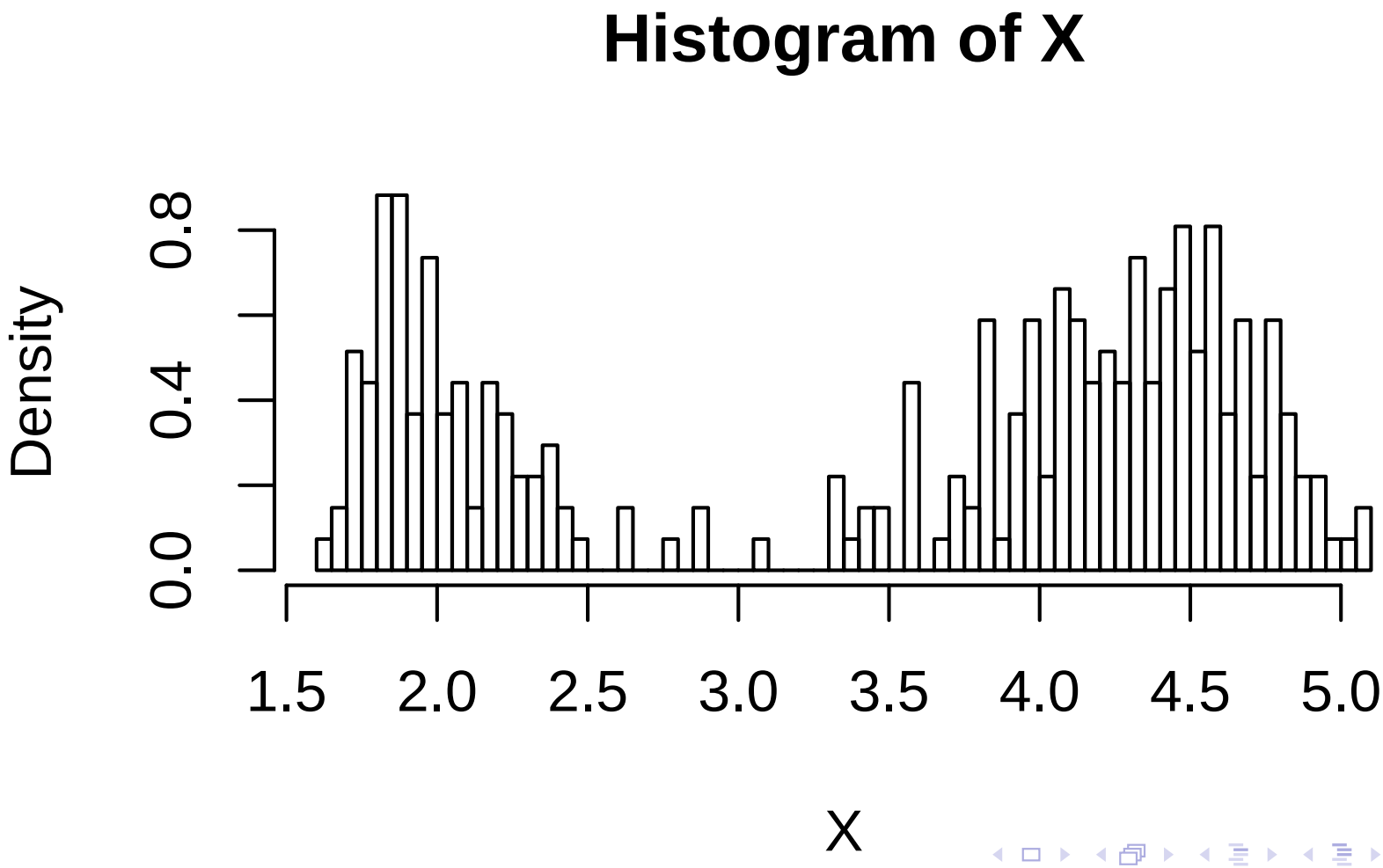
La distribution d'intérêt est donc bien la distribution invariante

Comparaison avec Metropolis-Hastings

- l'échantillonneur de Gibbs peut être vu comme un analogue de l'algorithme de Métropolis-Hastings
- la proposition concerne à tour de rôle chacune des dimensions de Θ
- la probabilité d'acceptation est systématiquement de 1
- il faut cependant être capable de déterminer (ou simuler suivant) les lois $\mathbb{P}(\theta_i | X, \Theta_{-i})$

Exemple : loi de mélange

```
> library(coda)
> data(faithful)
> X <- faithful$eruptions
> hist(X,breaks=100,freq=FALSE)
```



Exemple : loi de mélange

On considère un mélange de deux gaussiennes :

- pour tout individu i, on tire une classe $Z_i \in \{1, 2\}$ avec $\mathbb{P}(Z_i = j) = \alpha_j$.
- $X_i | Z_i = j \sim \mathcal{N}(\mu_j, \sigma_j^2)$

Remarque : Ce problème peut être considéré comme un problème à variable caché et être résolu par un algorithme EM.

Exemple : loi de mélange

- On pose $\theta_i = Z_i$.
- Il est difficile de simuler suivant $\mathbb{P}(\Theta|X)$.
- Par contre, sachant Θ_{-i}, on peut l'utiliser pour estimer (α_1, α_2), (μ_1, σ_1) et (μ_2, σ^2) puis simuler θ_i avec

$$\mathbb{P}(\theta_i = j) = \frac{\hat{\alpha_i} f(x_i, \hat{\mu_j}, \hat{\sigma_j})}{\hat{\alpha_1} f(x_i, \hat{\mu_1}, \hat{\sigma_1}) + \hat{\alpha_2} f(x_i, \hat{\mu_2}, \hat{\sigma_2})}$$

où $f(x, \mu, \sigma)$ désigne la densité en x de la loi $\mathcal{N}(\mu, \sigma)$.

Exemple : loi de mélange

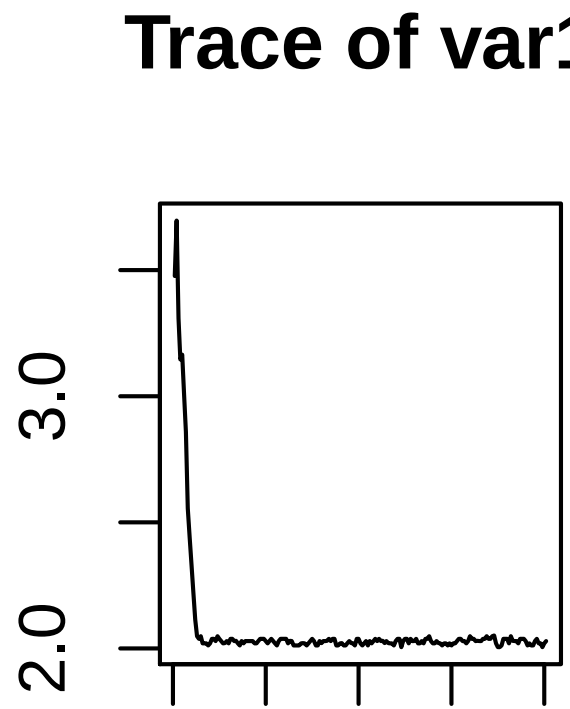
```
> singleupdate <- function(X,theta,i){
+   keep <- (c(1:length(theta))!=i) #toutes les coordonnées sauf i
+   sample1 <- X[keep & (theta==1)]  #échantillon des indices différent
+   sample2 <- X[keep & (theta==2)]  #échantillon des indices différent
+   mu1 <- mean(sample1)
+   sigma1 <- sd(sample1)
+   mu2 <- mean(sample2)
+   sigma2 <- sd(sample2)
+
+   proba1 <- dnorm(X[i],mu1,sigma1) / (dnorm(X[i],mu1,sigma1) + dnorm(
+
+   if (runif(1)<proba1 ){
+       theta[i]=1
+   } else {
+       theta[i]=2
+   }
+   return(theta)
+ }
```

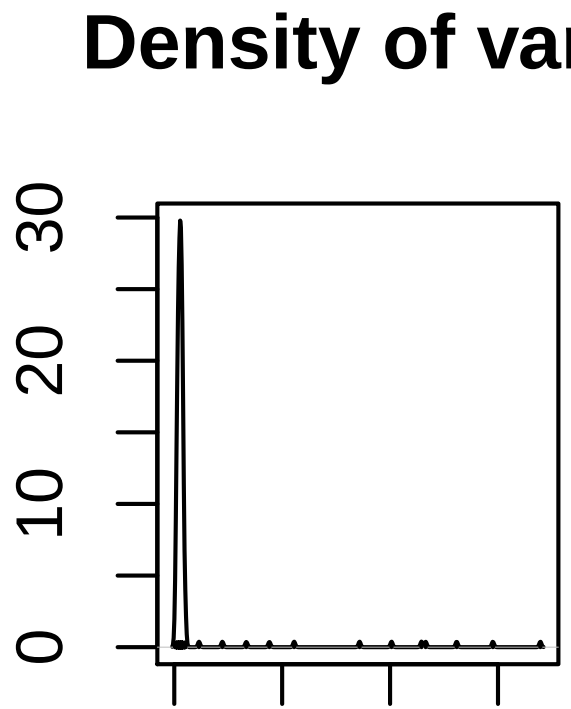
Exemple : loi de mélange

```
> globalupdate <- function(X,theta){
+   for (i in 1:length(theta)){
+     theta <- singleupdate(X,theta,i)
+   }
+   return(theta)
+ }
> mixturegibbs <- function(X,N){
+
+   theta <- sample(c(1,2),length(X),replace=TRUE)
+   mu1 <- mean(X[theta==1])
+   sigma1 <- sd(X[theta==1])
+   mu2 <- mean(X[theta==2])
+   sigma2 <- sd(X[theta==2])
+   alpha1 <- sum(theta==1)/length(X)
+
+   thetamatrix <- theta
+   mu1vect <- mu1
+   sigma1vect <- sigma1
+   mu2vect <- mu2
+   sigma2vect <- sigma2
+   alpha1vect <- alpha1
+
+
```

Exemple : loi de mélange

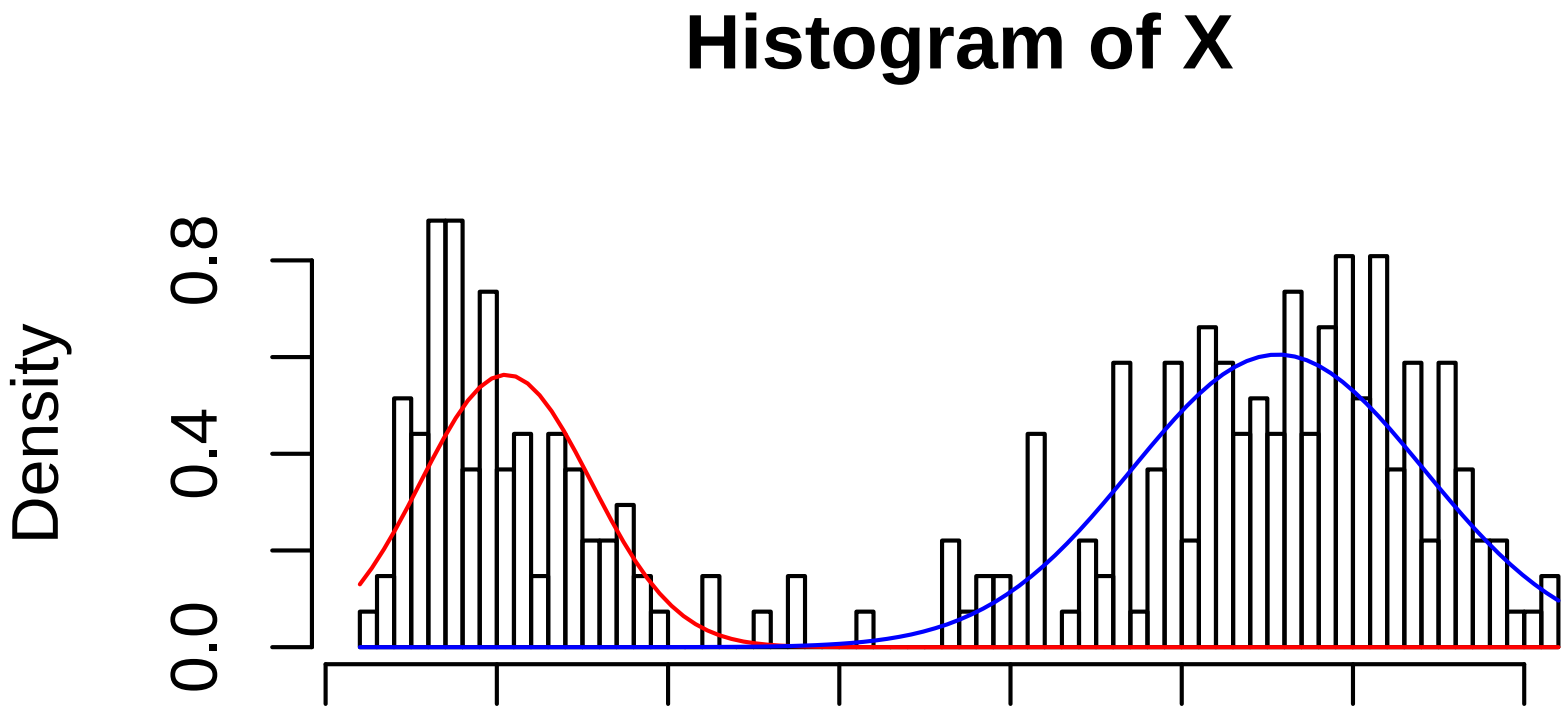
```
> res <- mixturegibbs(X,200)
> theta <- as.mcmc(res$theta)
> mu1 <- as.mcmc(res$mu1)
> mu2 <- as.mcmc(res$mu2)
> sigma1 <- as.mcmc(res$sigma1)
> sigma2 <- as.mcmc(res$sigma2)
> alpha1 <- as.mcmc(res$alpha1)
> plot(mu1)
```





Exemple : loi de mélange

```
> mu1est <- mean(mu1[50:200])
> mu2est <- mean(mu2[50:200])
> sigma1est <- mean(sigma1[50:200])
> sigma2est <- mean(sigma2[50:200])
> alpha1est <- mean(alpha1[50:200])
> hist(X,breaks=100,freq=FALSE)
> curve(alpha1est*dnorm(x,mu1est,sigma1est),add=TRUE,col='red')
> curve((1-alpha1est)*dnorm(x,mu2est,sigma2est),add=TRUE,col='blue')
```



IV.2 Statistiques bayésiennes et échantillonnage de Gibbs : notion de lois conjuguées

Rappel sur les statistiques bayésiennes

- On part d'une loi à priori du paramètre Θ d'intérêt et d'un observation X. Comment X modifie-t-elle la loi de Θ ?
- $$\mathbb{P}(\Theta|X) \propto \mathbb{P}(X|\Theta)\mathbb{P}(\Theta)$$
- La distribution obtenue est appelée loi à postériori de Θ.

Loi conjuguée

- Supposons que la vraisemblance $\mathbb{P}(X|\Theta)$ appartient à une famille $\mathcal{F}_1$ connue de lois paramétriques (normale, binomiale, ...).
- Une loi conjuguée est une famille de lois paramétriques $\mathcal{F}_2$ telle que si la loi à priori appartient à $\mathcal{F}_2$, la loi à postériori appartient également à $\mathcal{F}_2$.
- Passer de la loi à priori à la loi à postériori revient alors simplement à remettre à jour les paramètres de la loi.
- Suivant le cas, des formules closes peuvent être déterminées pour cela.

Lois conjuguées : Exemple

- Supposons que la vraisemblance suit une modèle de Poisson.

$$\mathbb{P}(\mathbf{x}\,|\theta) \propto \prod_i \left(\theta^{x_i} e^{-\theta}\right)$$

- Choisissons une loi Gamma comme loi à priori :

$$p(\theta) \propto \theta^{\alpha-1} e^{-\beta\theta}$$

- Alors la loi à priori est encore une loi Gamma

$$p(\theta|X) \propto \theta^{\alpha+\sum_i x_i-1} e^{-(\beta+n)\theta}$$